



US009530461B2

(12) **United States Patent**  
**Akers et al.**

(10) **Patent No.:** **US 9,530,461 B2**  
(45) **Date of Patent:** **Dec. 27, 2016**

(54) **ARCHITECTURES AND TECHNIQUES FOR PROVIDING LOW-POWER STORAGE MECHANISMS**

(75) Inventors: **Jason B. Akers**, Hillsboro, OR (US); **Knut S. Grimsrud**, Forest Grove, OR (US); **Robert J. Royer, Jr.**, Portland, OR (US); **Richard P. Mangold**, Forest Grove, OR (US); **Sanjeev Trika**, Portland, OR (US)

(73) Assignee: **Intel Corporation**, Santa Clara, CA (US)

(\* ) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 708 days.

(21) Appl. No.: **13/537,553**

(22) Filed: **Jun. 29, 2012**

(65) **Prior Publication Data**

US 2014/0003145 A1 Jan. 2, 2014

(51) **Int. Cl.**

**G11C 7/10** (2006.01)  
**G11C 5/14** (2006.01)  
**G06F 1/32** (2006.01)  
**G06F 11/14** (2006.01)

(52) **U.S. Cl.**

CPC ..... **G11C 5/148** (2013.01); **G06F 1/3287** (2013.01); **G06F 11/1417** (2013.01); **Y02B 60/1282** (2013.01)

(58) **Field of Classification Search**

CPC ..... G11C 14/00; G11C 7/00; H04W 52/0261; H04N 19/423

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,458,307 A \* 7/1984 McAnlis ..... G06F 11/1441 365/228  
2007/0094454 A1 \* 4/2007 Brundula ..... G06F 12/06 711/154  
2010/0073023 A1 \* 3/2010 Oh et al. .... 326/30  
2011/0087913 A1 \* 4/2011 Robles ..... G06F 1/3215 713/330  
2011/0125990 A1 \* 5/2011 Khosravi et al. .... 713/2  
2012/0125990 A1 \* 5/2012 Lin ..... B65D 3/06 229/404

\* cited by examiner

*Primary Examiner* — Son Dinh

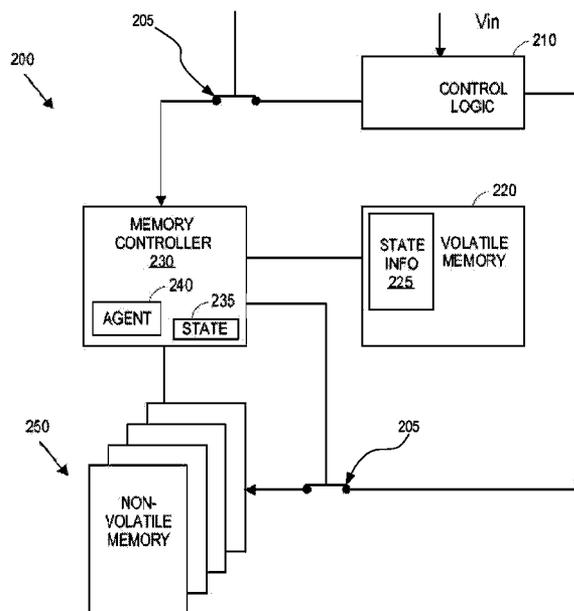
*Assistant Examiner* — Uyen B Tran

(74) *Attorney, Agent, or Firm* — Blakely, Sokoloff, Taylor & Zafman LLP

(57) **ABSTRACT**

Techniques to utilize a very low power state with a memory subsystem that includes one or more non-volatile memory devices and a volatile memory system. A memory controller is coupled with the one or more non-volatile memory devices and the volatile memory system. The memory controller comprising at least an embedded control agent and memory locations to store state information. The memory controller to selectively enable and disable the one or more non-volatile memory devices. The memory controller transfers the state information to the volatile memory system prior to entering a low power state. Control circuitry is coupled with the memory controller. The control circuitry to selectively enable and disable operation of the memory controller.

**23 Claims, 10 Drawing Sheets**



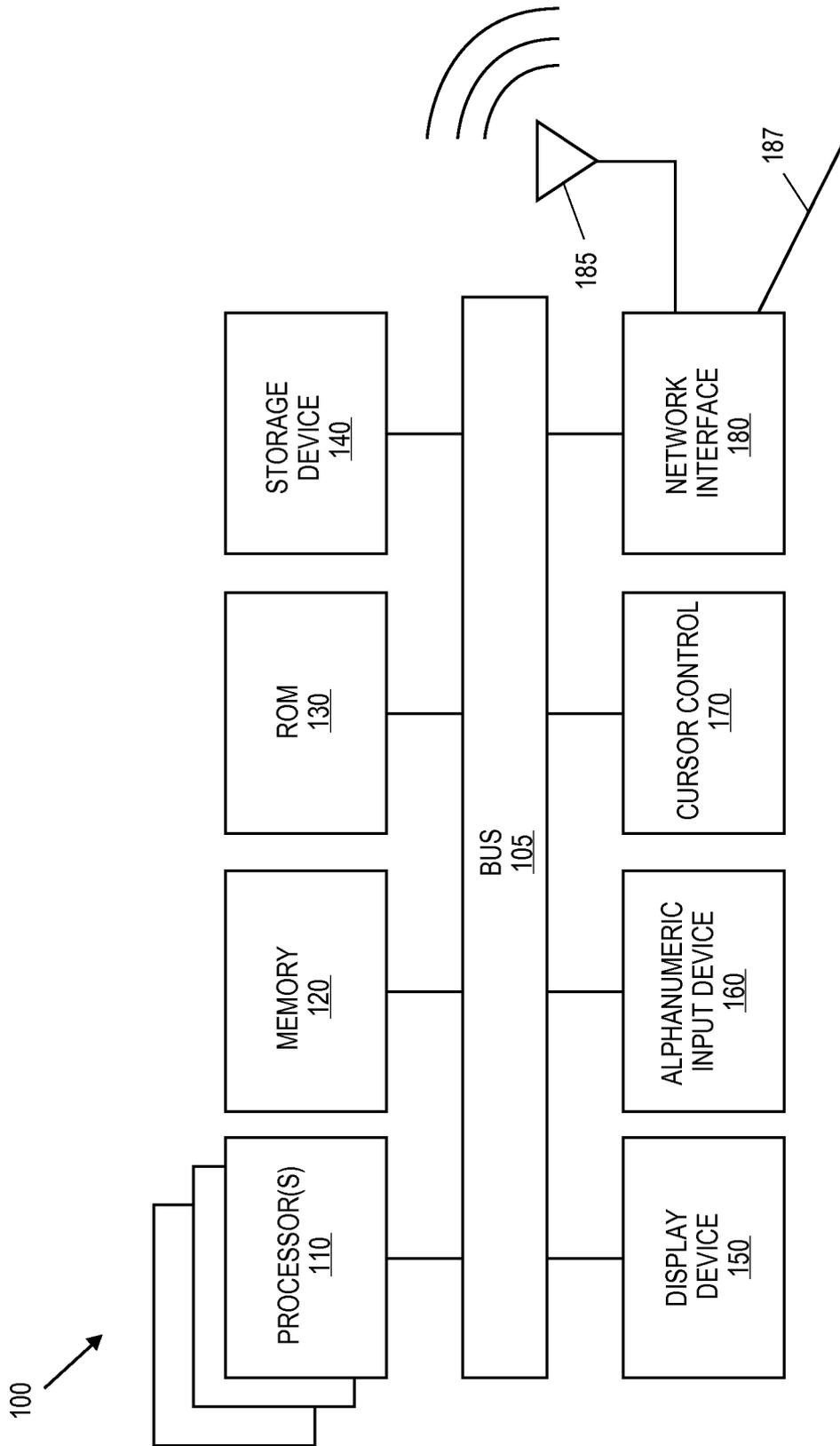


FIG. 1

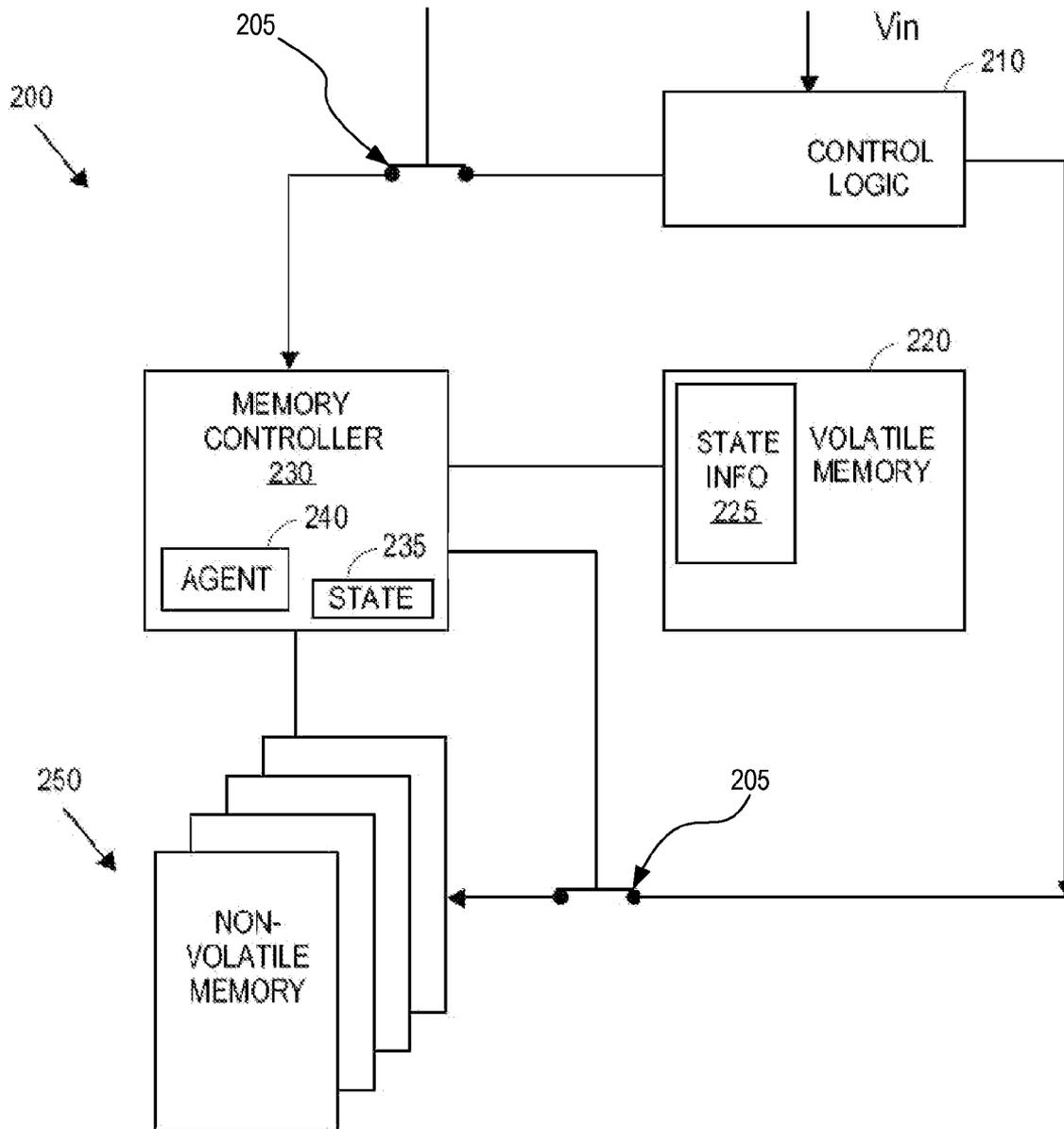


FIG. 2

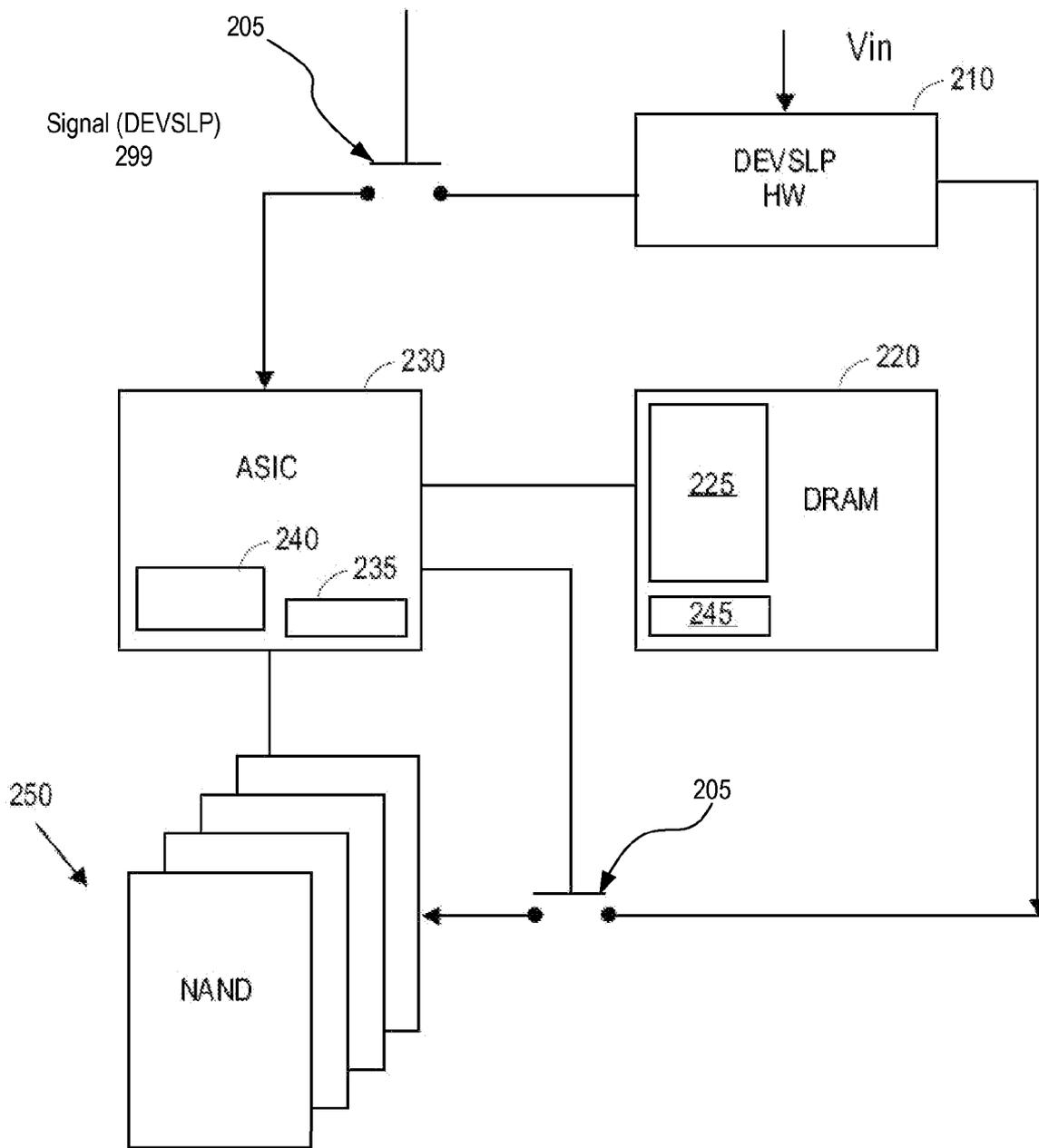


FIG. 3

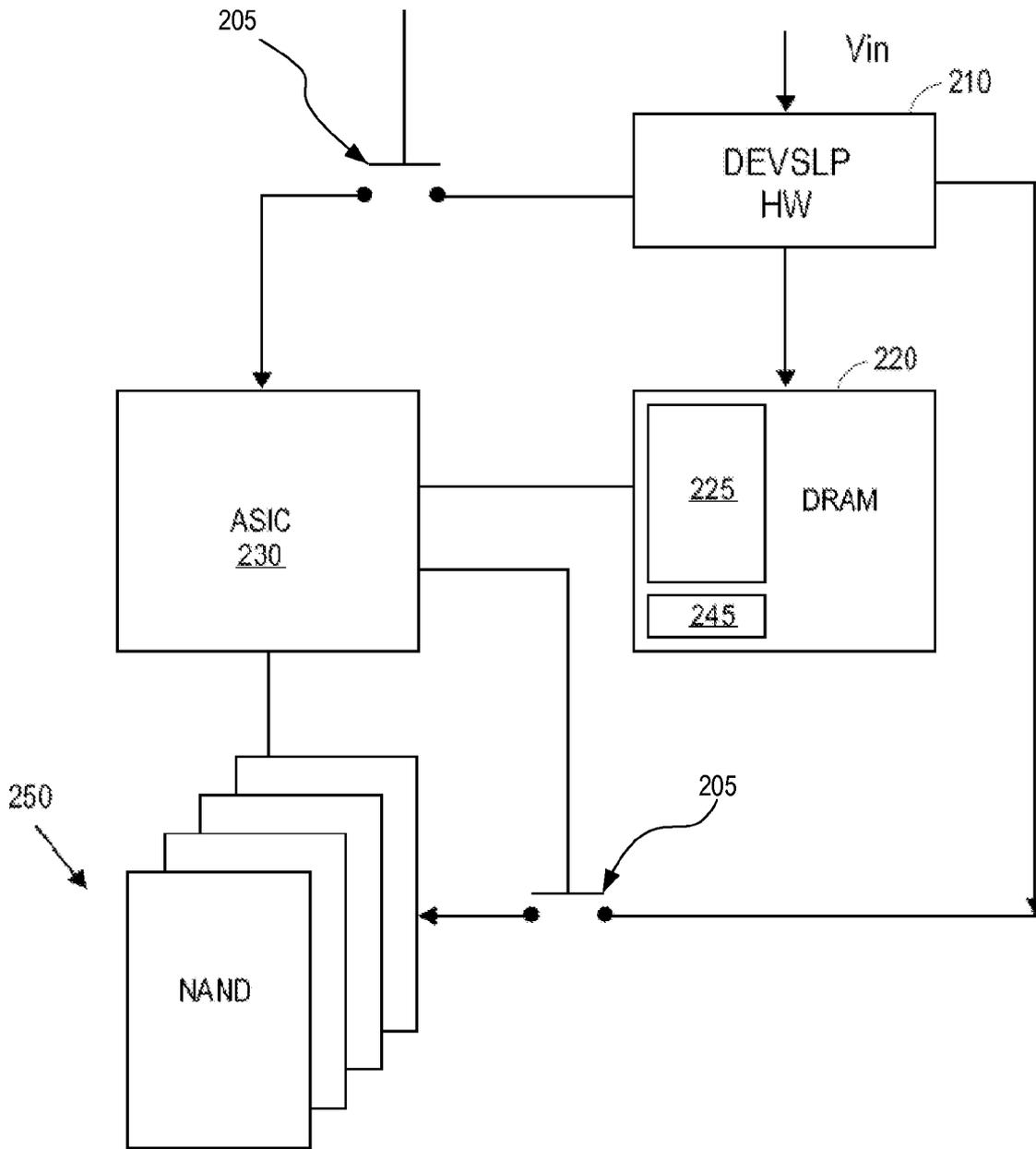


FIG. 4

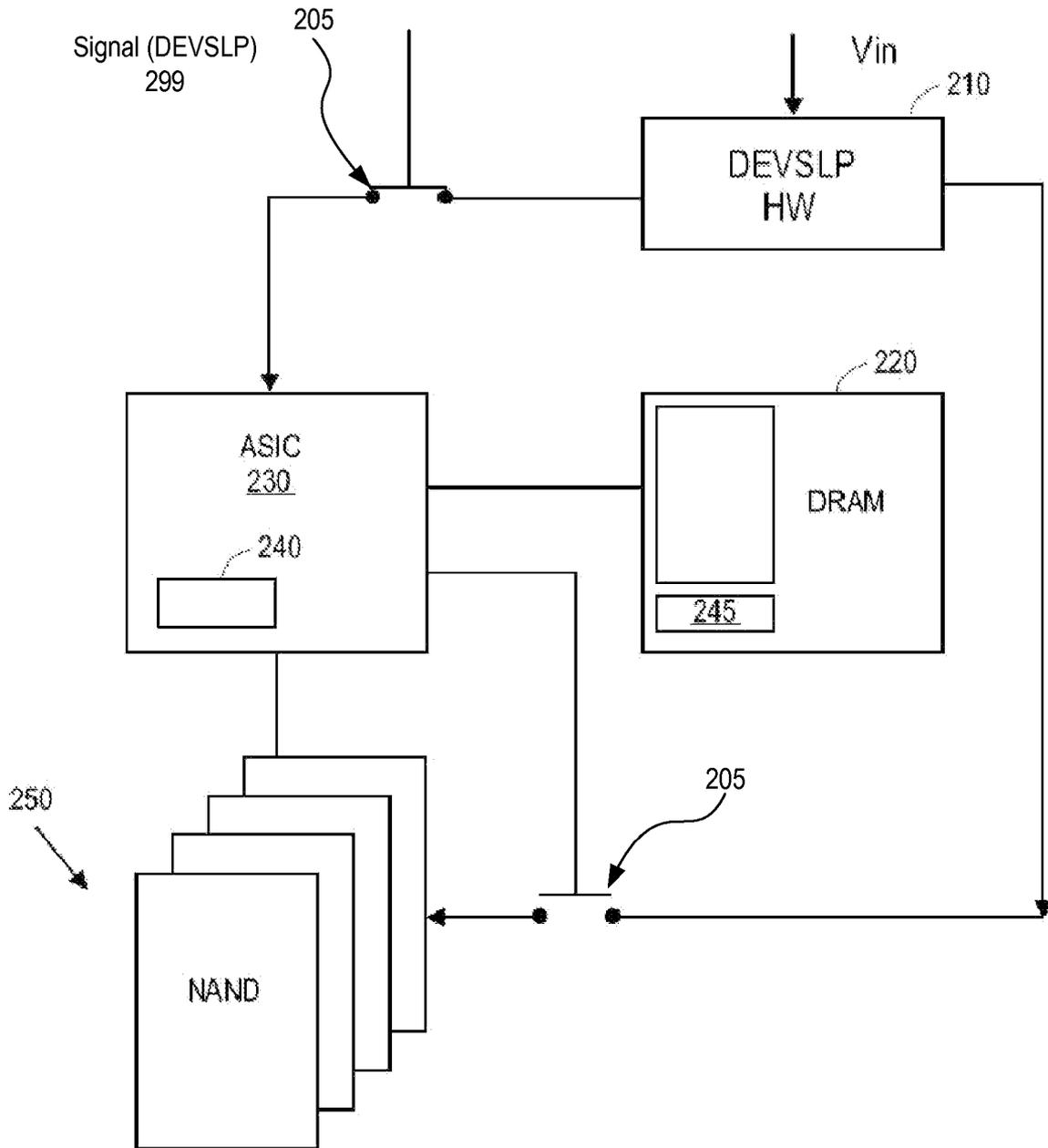


FIG. 5

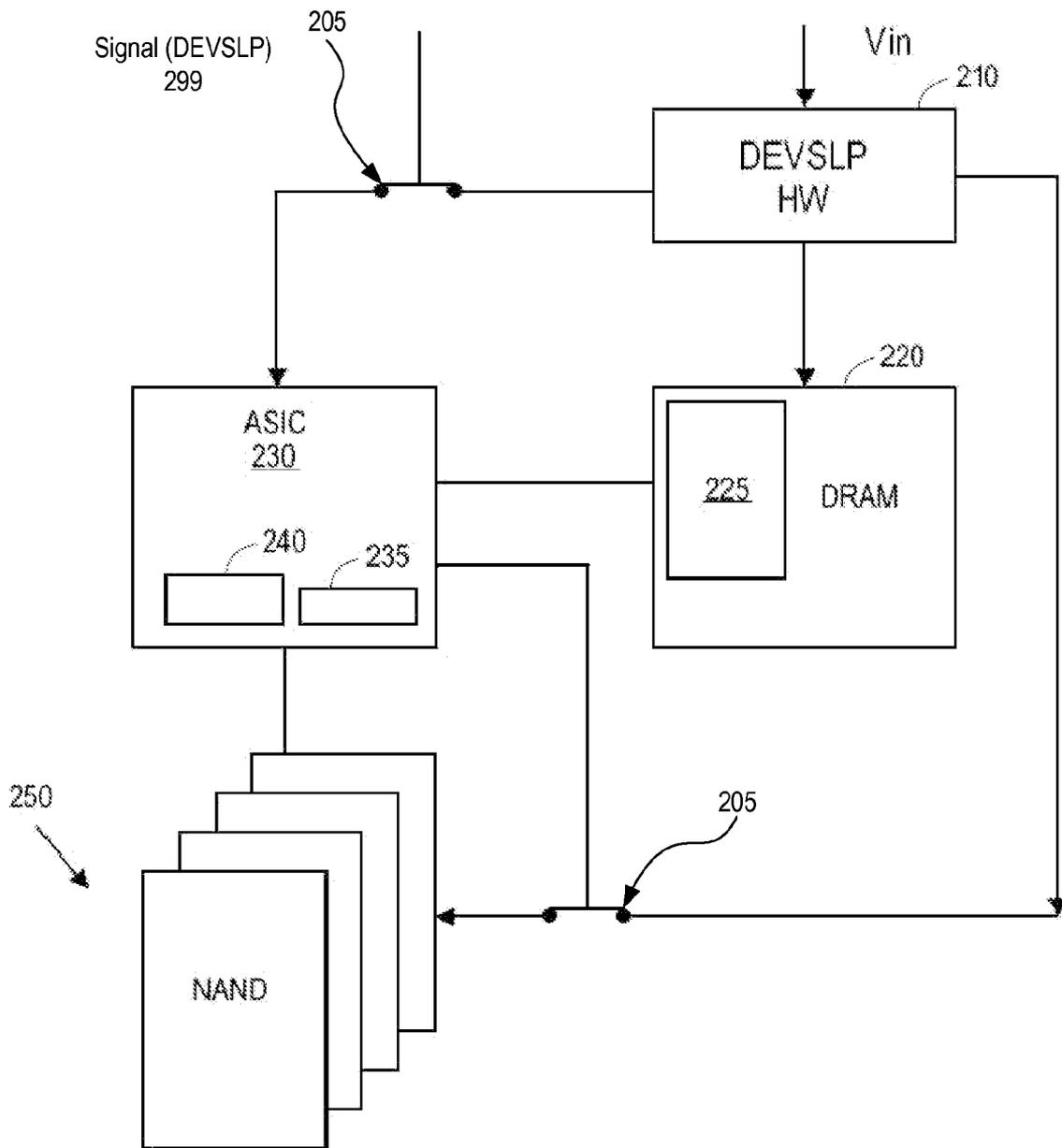
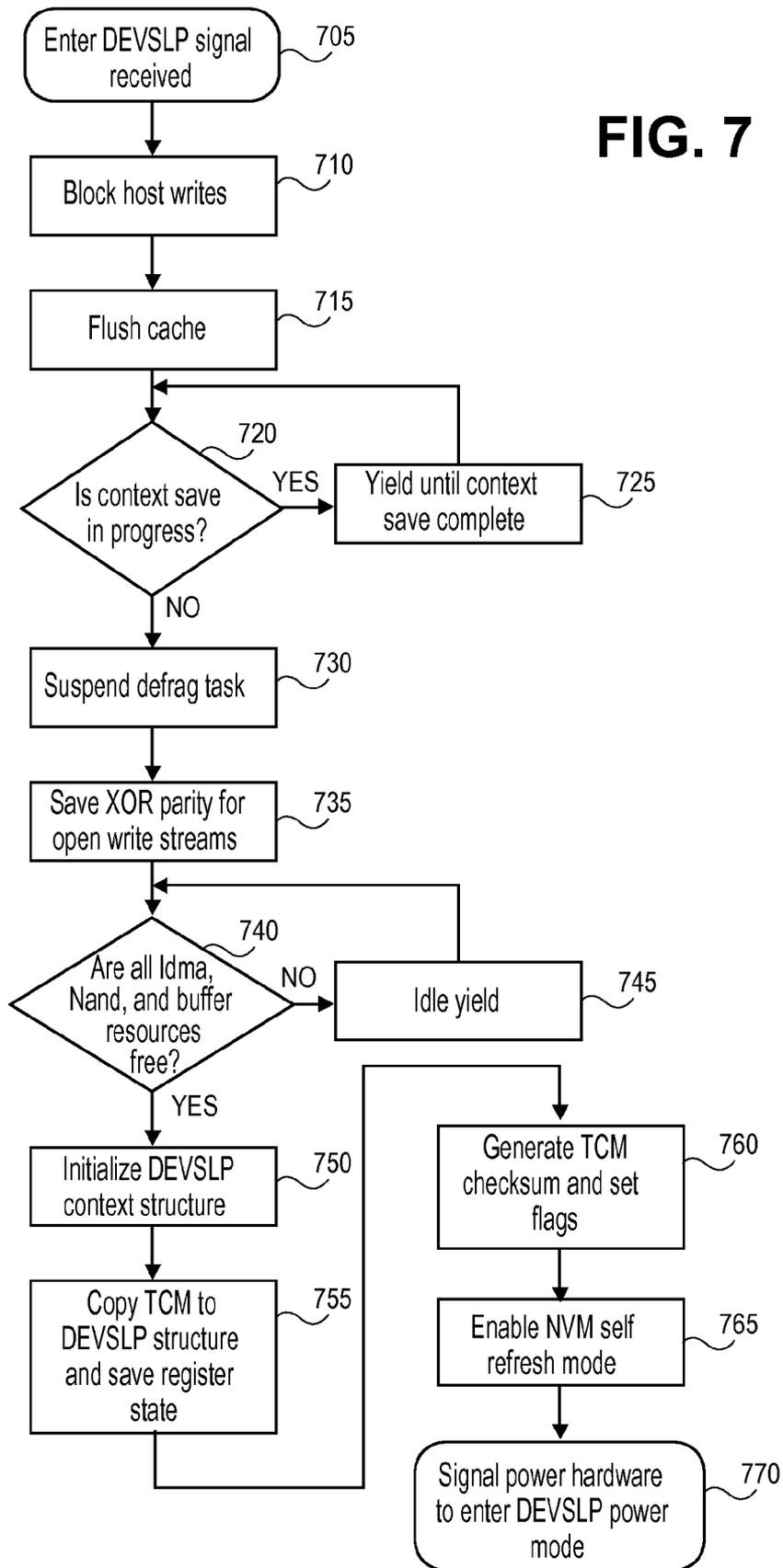


FIG. 6

FIG. 7



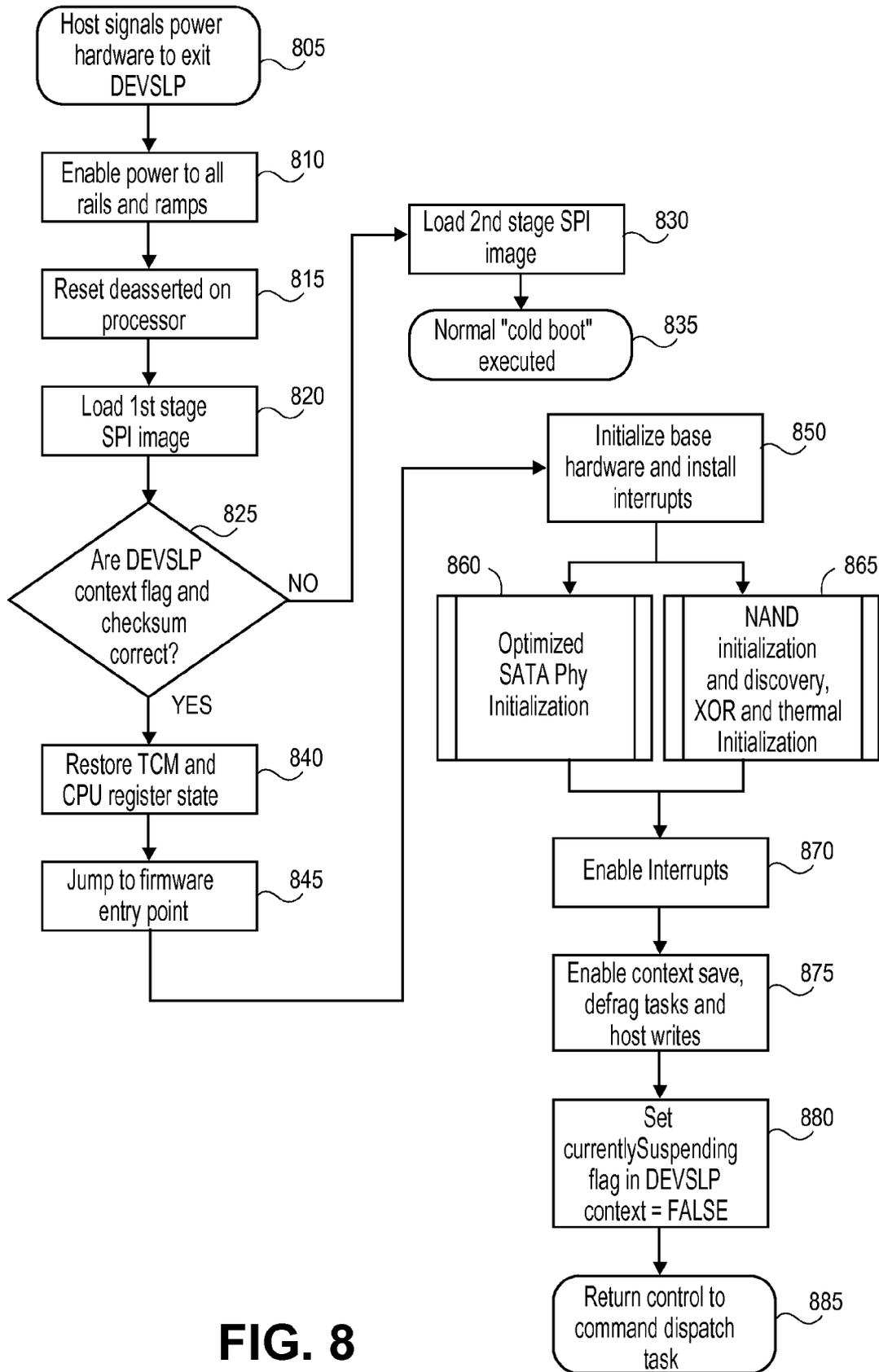


FIG. 8

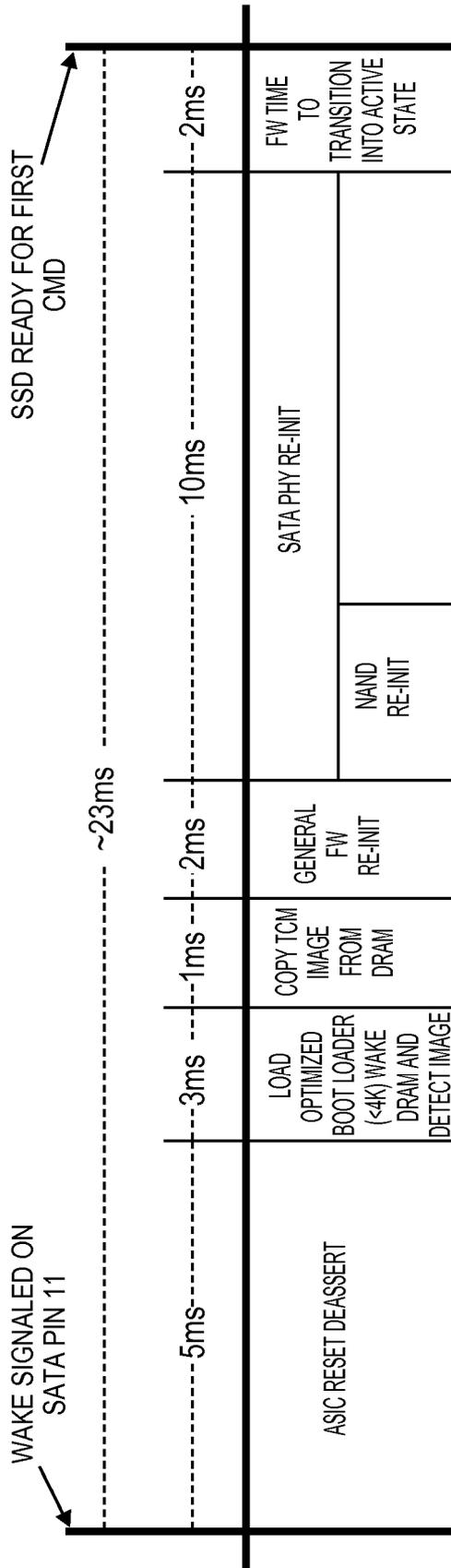
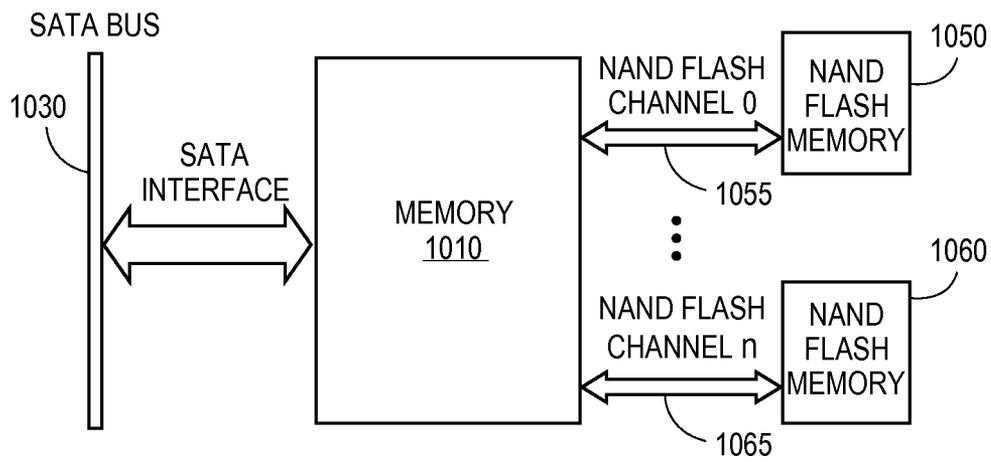


FIG. 9



**FIG. 10**

# ARCHITECTURES AND TECHNIQUES FOR PROVIDING LOW-POWER STORAGE MECHANISMS

## TECHNICAL FIELD

Embodiments of the invention relate to electronic devices. More particularly, embodiments of the invention relate to architectures and techniques for utilizing a low-power state in an electronic device, for example, a laptop computer system or tablet device.

## BACKGROUND

Electronic devices consume varying amounts of power during operation. For battery-powered devices, reduced power consumption can result in longer battery life. Thus, various techniques have been developed to reduce power consumption. However, most power-saving strategies sacrifice performance, which reduces user satisfaction.

## BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements.

FIG. 1 is a block diagram of one embodiment of an electronic system.

FIG. 2 is a block diagram of one embodiment of a memory subsystem that may utilize a deep sleep state as described herein.

FIG. 3 is a block diagram of one embodiment of a memory subsystem that may transition to a deep sleep state as described herein.

FIG. 4 is a block diagram of one embodiment of a memory subsystem that has transitioned to a deep sleep state as described herein.

FIG. 5 is a block diagram of one embodiment of a memory subsystem that may transition out of a deep sleep state as described herein.

FIG. 6 is a block diagram of one embodiment of a memory subsystem that may continue the transition out of a deep sleep state as described herein.

FIG. 7 is a flow diagram for one embodiment of entering a low power state.

FIG. 8 is a flow diagram for one embodiment of exiting a low power state.

FIG. 9 is a timing diagram corresponding to an exit from a low power state following one embodiment of the techniques described herein.

FIG. 10 is a block diagram of one embodiment of a solid state drive (SSD) system.

## DETAILED DESCRIPTION

In the following description, numerous specific details are set forth. However, embodiments of the invention may be practiced without these specific details. In other instances, well-known circuits, structures and techniques have not been shown in detail in order not to obscure the understanding of this description.

Described herein is a protocol for entering and exiting a very low power operational (DEVSLP) state, techniques for providing the state within a storage system, etc. In various embodiments, the very low power operational state func-

tions through a combination of a protocol, firmware (or other) functionality, and/or storage device hardware support.

A host system may cause entry into the very low power state by issuing, for example, a sleep Serial Advanced Technology Attachment (SATA) protocol command. In one embodiment an embedded agent (e.g., firmware) may respond by preparing a sleep image that may include, for example, memory controller state information, hardware state information, a buffer for storing bootloader instructions, a verification checksum, that may be stored in a dynamic memory device. The embedded agent may then cause the non-volatile memory to be powered down, or turned off, for example, by causing power to no longer be provided to the non-volatile memory.

FIG. 1 is a block diagram of one embodiment of an electronic system. The electronic system illustrated in FIG. 1 is intended to represent a range of electronic systems (either wired or wireless) including, for example, desktop computer systems, laptop computer systems, cellular telephones, personal digital assistants (PDAs) including cellular-enabled PDAs, set top boxes. Alternative electronic systems may include more, fewer and/or different components.

Electronic system 100 includes bus 105 or other communication device to communicate information, and processor 110 coupled to bus 105 that may process information. While electronic system 100 is illustrated with a single processor, electronic system 100 may include multiple processors and/or co-processors. Electronic system 100 further may include random access memory (RAM) or other dynamic storage device 120 (referred to as main memory), coupled to bus 105 and may store information and instructions that may be executed by processor 110. Main memory 120 may also be used to store temporary variables or other intermediate information during execution of instructions by processor 110.

Electronic system 100 may also include read only memory (ROM) and/or other static storage device 130 coupled to bus 105 that may store static information and instructions for processor 110. Data storage device 140 may be coupled to bus 105 to store information and instructions. Data storage device 140 such as a magnetic disk or optical disc and corresponding drive may be coupled to electronic system 100.

Electronic system 100 may also be coupled via bus 105 to display device 150, such as a cathode ray tube (CRT) or liquid crystal display (LCD), to display information to a user. Alphanumeric input device 160, including alphanumeric and other keys, may be coupled to bus 105 to communicate information and command selections to processor 110. Another type of user input device is cursor control 170, such as a mouse, a trackball, or cursor direction keys to communicate direction information and command selections to processor 110 and to control cursor movement on display 150.

Electronic system 100 further may include network interface(s) 180 to provide access to a network, such as a local area network. Network interface(s) 180 may include, for example, a wireless network interface having antenna 185, which may represent one or more antenna(e). Network interface(s) 180 may also include, for example, a wired network interface to communicate with remote devices via network cable 187, which may be, for example, an Ethernet cable, a coaxial cable, a fiber optic cable, a serial cable, or a parallel cable.

In one embodiment, network interface(s) 180 may provide access to a local area network, for example, by con-

forming to IEEE 802.11b and/or IEEE 802.11g standards, and/or the wireless network interface may provide access to a personal area network, for example, by conforming to Bluetooth standards. Other wireless network interfaces and/or protocols can also be supported.

IEEE 802.11b corresponds to IEEE Std. 802.11b-1999 entitled "Local and Metropolitan Area Networks, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band," approved Sep. 16, 1999 as well as related documents. IEEE 802.11g corresponds to IEEE Std. 802.11g-2003 entitled "Local and Metropolitan Area Networks, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 4: Further Higher Rate Extension in the 2.4 GHz Band," approved Jun. 27, 2003 as well as related documents. Bluetooth protocols are described in "Specification of the Bluetooth System: Core, Version 1.1," published Feb. 22, 2001 by the Bluetooth Special Interest Group, Inc. Associated as well as previous or subsequent versions of the Bluetooth standard may also be supported.

In addition to, or instead of, communication via wireless LAN standards, network interface(s) 180 may provide wireless communications using, for example, Time Division, Multiple Access (TDMA) protocols, Global System for Mobile Communications (GSM) protocols, Code Division, Multiple Access (CDMA) protocols, and/or any other type of wireless communications protocol.

FIG. 2 is a block diagram of one embodiment of a memory subsystem that may utilize a deep sleep state as described herein. The memory subsystem may be, for example, included within memory 120 of electronic system 100 as illustrated in FIG. 1, or within another type of electronic system.

FIG. 2 illustrates memory system 200 under normal operating conditions. Control logic 210 provides an enable signal to memory controller 230, volatile memory 220 and non-volatile memory 250. Volatile memory 220 may be, for example, dynamic random access memory (DRAM) or any other type of volatile memory. Volatile memory 220 may store state information for memory controller, 230, non-volatile memory 250 and/or other system components (not illustrated in FIG. 2).

In one embodiment, memory controller 230 includes agent 240 that operates to provide transitions 205 into and out of low power states as described herein. Agent 240 may be hardware, software, firmware or any combination thereof. In one embodiment, memory controller 230 also stores state 235, which is state information corresponding to non-volatile memory 250. Memory controller 230 may also cause state information 225 to be stored in volatile memory 220.

Thus, in the conditions illustrated in FIG. 2, each of the components of system 200 are consuming normal amounts of power, which can vary depending on the types of devices, operating environment, etc. In order to extend battery life and/or conserve power consumption, it may be desirable to reduce power consumption of as many devices as possible.

FIG. 3 is a block diagram of one embodiment of a memory subsystem that may transition 205 to a deep sleep state as described herein. FIG. 3 illustrates conditions during a transition from normal operating mode to a reduced power state.

In the example of FIG. 3, power has been removed from non-volatile memory 250 by memory controller 230 in response to a signal 299 (e.g., DEVSLP) from a host system (not illustrated in FIG. 3). In one embodiment, agent 240 operates to cause memory controller 230 to turn off power

to non-volatile memory 250. Further, agent 240 causes state information 235 to be copied to volatile memory and be stored as state information 245. Agent 240 and/or memory controller 230 may also cause volatile memory 220 to enter a low power state.

FIG. 4 is a block diagram of one embodiment of a memory subsystem that has transitioned 205 to a deep sleep state as described herein. After volatile memory 220 is placed in a low power state and non-volatile memory 250 is turned off, agent 240 may cause memory controller 230 to enter a low power or no power state.

In this state, only control logic 210 and non-volatile memory 250 receive power. In one embodiment, volatile memory 220 receives only reduced power in this state. This allows system 200 to be in a lower power state than traditionally available through placing memory components in low power states.

FIG. 5 is a block diagram of one embodiment of a memory subsystem that may transition 205 out of a deep sleep state as described herein. In response to the signal 299 (e.g., DEVSLP) state from the host system changing, control logic 210 turns memory controller 230 back on and agent 240 begins operating again.

FIG. 6 is a block diagram of one embodiment of a memory subsystem that may continue the transition 205 out of a deep sleep state as described herein. In one embodiment, agent 240 operates to restore state information 245 to memory controller 230, which allows memory controller 230 to avoid the traditional step of rebuilding state as required by traditional mechanisms when turning non-volatile memory back on. This may save considerable time and provide a better operating experience utilizing the low power state described herein.

Memory controller 230 and/or agent 240 operate to turn on non-volatile memory 250 as well as other hardware components. This restores the system illustrated in FIGS. 2-5 to normal operating state where both volatile memory 220 and non-volatile memory 250 are accessible via memory controller 230.

FIG. 7 is a flow diagram for one embodiment of entering a low power state. The technique of FIG. 7 may be used, for example, with the components illustrated in FIGS. 2-6. One example is provided in FIG. 7 and one or more operations described herein may be accomplished in a different order and some steps may be optional.

A host system generates/asserts a signal to enter the sleep (or low power) state ("DEVSLP"), 705. This signal may be generated/asserted in any manner known in the art. For example, the signal may be under the control of a host operating system that monitors system conditions. As another example, the signal may respond to a hard input, such as a button, to allow a user to cause the system to enter the low power state. Other options can also be supported.

Writes from the host system are blocked, 710. This can be accomplished by memory controller 230, or by the memory controller in response to a signal from agent 240. One or more caches may be flushed 715, depending on the cache structure being utilized in the system. This can be accomplished by memory controller 230, or by the memory controller in response to a signal from agent 240.

If a context save is in progress, 720, the process yields until the context save is complete, 725. If a context save is not in progress, 720, any defragmentation tasks are suspended, 730 and parity information for open write streams can be saved, 735.

If direct memory access, non-volatile memory and buffer resources are not free, 740, the process will idle, 745. If

direct memory access, non-volatile memory and buffer resources are free, **740**, a context structure is initialized, **750**. Memory controller state information is copied to the context structure, which is saved along with register state information, **755**.

In one embodiment, a checksum is generated for the context information and associated flags are set, **760**. Non-volatile memory self-refresh mode is enabled, **765** and the remaining hardware (e.g., memory controller, non-volatile memory) enter a very low power, or no power mode, **770**.

FIG. **8** is a flow diagram for one embodiment of exiting a low power state. The technique of FIG. **8** may be used, for example, with the components illustrated in FIGS. **2-6**.

A host system generates/asserts a signal to exit the sleep (or low power) state, **805**. This signal may be generated/asserted in any manner known in the art. For example, the signal may be under the control of a host operating system that monitors system conditions. As another example, the signal may respond to a hard input, such as a button, to allow a user to cause the system to enter the low power state. Other options can also be supported.

Power is enabled to the power distribution network, **810**. In one embodiment, a reset signal is deasserted on a processor, **815**. The first stage of the sleep image is loaded, **820**. If the checksums for the sleep image are incorrect, **825**, a second stage of the image is loaded, **830** and a cold boot is executed, **835**. If the checksums are correct, **825**, the memory controller and register state are restored, **840** and execution may jump to a firmware entry point, **845**.

Base hardware elements are initialized and interrupts are installed, **850**. In one embodiment, an optimized physical layer initialization is performed, **860**, and non-volatile memory is initialized, **865**. Interrupts are enabled, **870**. Context save, defragmentation tasks and host write functionality are enabled, **875**. The system is marked as not being in the deep sleep (DEVSLP) mode, **880**. Control is returned to the task, **885**.

Techniques to utilize a very low power state with a memory subsystem that includes one or more non-volatile memory devices and a volatile memory system. A memory controller is coupled with the one or more non-volatile memory devices and the volatile memory system. The memory controller comprising at least an embedded control agent and memory locations to store state information. The memory controller to selectively enable and disable the one or more non-volatile memory devices. The memory controller transfers the state information to the volatile memory system prior to entering a low power state. Control circuitry is coupled with the memory controller. The control circuitry to selectively enable and disable operation of the memory controller.

In one embodiment, the embedded control agent comprises firmware within the memory controller. In response to a signal from the control circuitry, the firmware causes a state corresponding to the memory controller to be copied to the volatile memory system, causes the one or more non-volatile memory devices to be turned off, and causes the volatile memory system to enter a low power state. In one embodiment, in response to a change in the signal from the control circuitry, the memory controller is enabled.

In one embodiment, in response to the change in the signal from the control circuitry, the firmware causes the state information from the non-volatile memory to be restored to the memory controller. In one embodiment, in response to the change in the signal from the control circuitry, the firmware causes the one or more non-volatile memory devices to be turned on.

The one or more non-volatile memory devices can be at least one NAND flash memory device. In one embodiment, the non-volatile memory subsystem comprises dynamic random access memory (DRAM). In one embodiment, the DRAM is at least compliant with one or more DDR-3 or later standards.

FIG. **9** is a timing diagram corresponding to an exit from a low power state following one embodiment of the techniques described herein. The timing diagram for operation of a solid state drive (SSD).

In one embodiment, a host system indicates entry into a low power (e.g., DevSleep) mode by issuing a sleep Serial Advanced Technology Attachment (SATA) command or by modifying a general purpose input/output (GPIO) register through chipset support. The firmware responds by preparing the PhySleep image as described above. The firmware then signals the DevSleep gate to power off.

When a Wake signal is asserted (e.g., on SATA pin **11**), the wake process described herein begins. In one embodiment, takes approximately 23 ms; however, other configurations can have other timing. The timing illustrated in FIG. **9** conforms with SATA international Organization (SATA-IO) guidelines for example, Serial ATA Specification, Revision 3.0 (released June 2009) or subsequent specifications available from SATA-IO Administration in Beaverton, Oreg.

During the first 5 ms, the ASIC reset signal is deasserted. During the following 3 ms, the optimized boot load is loaded, the DRAM wakes and the image is detected. The image is loaded from the DRAM in the following 1 ms and the general firmware is reinitialized in the following 2 ms. The subsequent 10 ms are used for reinitialization of the SATA physical link and reinitialization of the SSD NAND memory. During the final 2 ms, the firmware transitions to the active state.

FIG. **10** is a block diagram of one embodiment of a solid state drive (SSD) system. The SSD of FIG. **10** can be a part of, for example, storage device **140** as illustrated in FIG. **1**. The memory system of FIG. **10** may be utilized in other systems as well. Current SSD devices include both volatile and non-volatile memory (e.g., NAND) and use a logical-to-physical (L2P) indirection system (e.g., a lookup table) to store and retrieve user data. This L2P information and other drive state information together may be referred to as "context." The context can be maintained in volatile memory and may be periodically backed up to non-volatile memory.

Memory controller **1010** operates to manage the flow of data between one or more memory devices and a data source (not illustrated in FIG. **10**). The data source may be, for example, a processor or other system component that may generate data or utilize data stored in memory. In one embodiment, memory controller **1010** communicates with SATA bus **1030** via SATA interface **1020** that provides an interface between memory controller **1010** and the data source.

In one embodiment, memory controller **1010** is a SATA-to-NAND controller that manages bandwidth between SATA bus **1030** and multiple NAND flash memory devices (e.g., **1050**, **1060**) over multiple NAND flash channels (e.g., **1055**, **1065**).

Reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

While the invention has been described in terms of several embodiments, those skilled in the art will recognize that the invention is not limited to the embodiments described, but can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of limiting.

What is claimed is:

1. An apparatus comprising:
  - one or more non-volatile memory devices;
  - a volatile memory system;
  - a memory controller coupled with the one or more non-volatile memory devices and the volatile memory system;
  - the memory controller having therein at least an embedded control agent and memory locations to store state information;
  - wherein the memory controller is to turn off power to the one or more non-volatile memory devices;
  - wherein the memory controller is to copy the state information to the volatile memory system prior to entering a low power state and further cause the volatile memory system to enter the low power state;
  - wherein the embedded control agent is to transition the apparatus out of a deep sleep mode by restoring the state information previously copied to the volatile memory system to the memory controller bypassing a rebuilding state when turning the one or more non-volatile memory devices back on after the memory controller turned off power to the one or more non-volatile memory devices; and
  - control circuitry coupled with the memory controller, the control circuitry to selectively enable and disable operation of the memory controller.
2. The apparatus of claim 1 wherein the memory controller to turn off power to the one or more non-volatile memory devices comprises the memory controller to selectively disable the one or more non-volatile memory devices and further wherein the memory controller to selectively enable the one or more non-volatile memory devices.
3. The apparatus of claim 1;
  - wherein the embedded control agent comprises firmware within the memory controller; and
  - wherein, in response to a signal from the control circuitry, the firmware causes the memory controller to copy the state information to the volatile memory system prior to entering the low power state and wherein the firmware causes the one or more non-volatile memory devices to be turned off, and wherein the firmware causes the volatile memory system to enter the low power state.
4. The apparatus of claim 3 wherein, in response to a change in the signal from the control circuitry, the memory controller is enabled.
5. The apparatus of claim 4 wherein, in response to the change in the signal from the control circuitry, the firmware causes the state information previously copied to the volatile memory system to be restored to the memory controller.
6. The apparatus of claim 5 wherein, in response to the change in the signal from the control circuitry, the firmware causes the one or more non-volatile memory devices to be turned on.
7. The apparatus of claim 1 wherein the one or more non-volatile memory devices comprise at least one NAND flash memory device.
8. The apparatus of claim 1 wherein the one or more non-volatile memory devices comprise at least a solid state drive (SSD).

9. The apparatus of claim 1 wherein the non-volatile memory devices comprise dynamic random access memory (DRAM).

10. The apparatus of claim 9 wherein the DRAM is at least compliant with one or more DDR-3 or later standards.

11. A method for causing a memory system to utilize a deep sleep state, the memory system comprising a memory controller coupled with one or more non-volatile memory devices, a volatile memory device and a control circuit, the method comprising:

- receiving, with an embedded control agent within the memory controller, a signal from the control circuit to initiate transition to the deep sleep state;
- causing state information for the memory controller to be stored in the volatile memory device prior to entering a low power mode;
- causing, via the memory controller, power to be turned off for the one or more non-volatile memory devices;
- causing, via the memory controller, the volatile memory device to enter the low power mode;
- transitioning, via the embedded control agent, the memory system out of the deep sleep state by restoring the state information previously copied to the volatile memory system to the memory controller bypassing a rebuilding state when turning the one or more non-volatile memory devices back on after the memory controller turned off power to the one or more non-volatile memory devices; and
- causing, via control circuitry of the memory system, the memory controller to enter the low power mode.

12. The method of claim 11 wherein the embedded control agent comprises firmware within the memory controller.

13. The method of claim 11 further comprising:
 

- receiving, with the embedded control agent within the memory controller, a signal from the control circuit to initiate transition out of the deep sleep state;
- transitioning the memory controller out of the low power mode;
- restoring, from the volatile memory device, the state information for the memory controller;
- transitioning the one or more non-volatile memory devices to an active state.

14. The method of claim 11 wherein the one or more non-volatile memory devices comprise at least one NAND flash memory device.

15. The method of claim 11 wherein the one or more non-volatile memory devices comprise at least one solid state drive (SSD).

16. The method of claim 11 wherein the non-volatile memory devices comprise dynamic random access memory (DRAM).

17. The method of claim 16 wherein the DRAM is at least compliant with one or more DDR-3 or later standards.

18. A system comprising:
 

- one or more non-volatile memory devices;
- a dynamic random access memory (DRAM) system;
- a memory controller coupled with the one or more non-volatile memory devices and the DRAM system;
- the memory controller having therein at least an embedded control agent and memory locations to store state information;
- wherein the memory controller is to turn off power to the one or more non-volatile memory devices;
- wherein the memory controller is to copy the state information to the DRAM system prior to entering a low power state and further cause a volatile memory system to enter the low power state;

wherein the embedded control agent is to transition the system out of a deep sleep mode by restoring the state information previously copied to the DRAM system to the memory controller bypassing a rebuilding state when turning the one or more non-volatile memory devices back on after the memory controller turned off power to the one or more non-volatile memory devices; and  
 control circuitry coupled with the memory controller, the control circuitry to selectively enable and disable operation of the memory controller.

19. The system of claim 18 wherein the memory controller to turn off power to the one or more non-volatile memory devices comprises the memory controller to selectively disable the one or more non-volatile memory devices and further wherein the memory controller to selectively enable the one or more non-volatile memory devices.

20. The system of claim 18:  
 wherein the embedded control agent comprises firmware within the memory controller; and

wherein, in response to a signal from the control circuitry, the firmware causes the memory controller to copy the state information to the volatile memory system prior to entering the low power state and wherein the firmware causes the one or more non-volatile memory devices to be turned off, and wherein the firmware causes the DRAM system to enter the low power state.

21. The system of claim 20 wherein, in response to a change in the signal from the control circuitry, the memory controller is enabled.

22. The system of claim 21 wherein, in response to the change in the signal from the control circuitry, the firmware causes the state information previously copied to the volatile memory system to be restored to the memory controller.

23. The system of claim 22 wherein, in response to the change in the signal from the control circuitry, the firmware causes the one or more non-volatile memory devices to be turned on.

\* \* \* \* \*